

## МЕТОДЫ ОЦЕНКИ ПОЛЬЗОВАТЕЛЬСКОГО ИНТЕРФЕЙСА. РАЗРАБОТКА ЭКСПЕРТНОЙ СИСТЕМЫ ОЦЕНКИ ИНТЕРФЕЙСА ПОСРЕДСТВОМ ОБОЛОЧКИ CLIPS

### Подходы проектирования пользовательского интерфейса

Пригодность к использованию и удобство пользовательского интерфейса изучается при помощи специальных методов, изучаемых в юзабилити (usability). Юзабилити - это качественный признак, который определяет, насколько интерфейс пользователя легок в использовании. Рассмотрим некоторые из этих методов. Эвристическая оценка (heuristic evaluation) это один из методов юзабилити, который служит для обнаружения проблем в дизайне пользовательского интерфейса. Эвристическая оценка проводится небольшой группой людей, которые оценивают интерфейс и судят о его правильности, опираясь на определенные общепризнанные принципы юзабилити (которые называются «эвристиками»). «Эвристики» были разработаны Якобом Нильсеном совместно с Рольфом Моличем. Это десять главных положений любого разработчика компьютерных интерфейсов, т. е. минимальные критерии, которым должен отвечать интерфейс любой программы. В целом эвристическая оценка - весьма трудоемкий процесс для одного человека, так как один человек никогда не сможет обнаружить все проблемы в интерфейсе, поэтому обычно участвуют в эвристической оценке от трех до пяти участников, так как при слишком большом количестве участников вы будете в основном получать повторяющуюся информацию.

Метод, основанный на проверке по контрольному списку, ближе всего к формальному тестированию. Составляется список произвольных требований, после чего интерфейс проверяется на соответствие этим требованиям. В контрольном списке может быть все, что угодно; как правило, используются требования из стандартов и руководств по проектированию интерфейсов. К сожалению, он имеет два принципиальных недостатка: чем более полон и детален контрольный список, тем дольше производится проверка; контрольный список, чтобы быть совершенно надежным, не имеет права содержать сколь угодно размытых требований. Кроме того, качество проверки по контрольному списку напрямую зависит от качества самого списка. Чтобы создать качественный список требований, нужно затратить годы труда – и самое главное – что этот список никогда не будет завершен, поскольку все время проявляются новые требования. С другой стороны, четкий контрольный список может использоваться кем угодно, даже непрофессионалами, в проектировании интерфейсов – как-никак вся необходимая информация уже содержится в списке.

Экспертная же оценка (expert evaluation) очень похожа на эвристическую оценку. Как правило, в оценке системы принимают участие два-три аналитика, которые проверяют дизайн системы на соответствие оговоренным правилам и

принципам. Они записывают свои наблюдения и сортируют их в порядке серьезности обнаруженных проблем. В качестве аналитиков приглашаются специалисты в области взаимодействия человека и компьютера или специалисты по человеческому фактору, но, как показывает опыт, и менее опытные специалисты были способны обнаружить серьезные проблемы.

### **Решение – разработка экспертной системы**

Качество экспертной оценки может быть сильно улучшено, если увеличить количество экспертов, следовательно, потребуются дополнительные затраты. Создание экспертной системы на основе знаний, опыта специалиста по юзабилити позволит решить данную проблему. Использование экспертной системы упростит работу при проектировании системы, поскольку данная система сможет предоставлять рекомендации, советы при разработке и давать оценку уже созданного приложения.

При разработке ЭС источниками знания могут выступать: принципы проектирования пользовательских интерфейсов («эвристики», практические принципы построения интерфейсов); специалисты по юзабилити; стандарты и требования проектирования пользовательских интерфейсов, рекомендации (ГОСТ Р ИСО/МЭК 12119-2000, ГОСТ Р ИСО/МЭК 9126-93, ИСО/МЭК 2382-9-95 ГОСТ 28806-90, ISO 9241 ISO 9241-11).

При проектировании экспертной системы можно использовать средство разработки экспертных систем CLIPS. CLIPS предоставляет три механизма представления знаний: правила, которые, прежде всего, предназначены для эвристических знаний, основанных на опыте (эвристический); deffunctions и общие функции, которые предназначены для процедурного знания (процедурный); объектно-ориентированное программирование, также прежде всего предназначено для процедурного знания (объектно-ориентированный).

Один из первичных методов представления знания в CLIPS – правило. Правила используются для представления эвристики, или «правила бегунка», которые определяют набор действий, которые нужно выполнить для данной ситуации. Правило составлено из antecedenta и следствия. Основными компонентами языка описания правил являются база фактов (fact base) и база правил (rule base). На них возлагаются следующие функции: база фактов представляет исходное состояние проблемы; база правил содержит операторы, которые преобразуют состояние проблемы, приводя его к решению.

Факты - одна из базисных форм высокого уровня для представления информации в системе CLIPS. Каждый факт представляет фрагмент информации, которая была помещена в текущий список фактов, названный .. список фактов –называемый fact-list. Факты - основная единица данных, используемых по правилам. Факты могут быть добавлены к базе фактов (использование утверждающей команды assert), удалены из базы фактов (использование отзывающей команды clear), изменены (использование изменяющей команды retract) или дублированы (использование дублирующей команды) либо пользователем, либо исполнением программы CLIPS.

Например:

```
defrule otv_w
```

```
(value ?n)
(test (and (>= ?n 4) (<= ?n 6)))
=>
(printout t "The this interface requires essential completions" crlf)
(printout t " " crlf)
```

В данном примере, если значение переменной *n* находит в интервале от 4 до 6, то на экран выводится сообщение "The this interface requires essential completions".

Правило активизируется, если

- 1) совершенно новый факт, который не существовал прежде, или
- 2) факт, который существовал прежде, но был удален и переподтвержден, то есть "аналог" старого объекта образца, и, таким образом, теперь есть новый объект образца.

Правило и индексы фактов являются активацией. Если правило или объект образца, или оба изменились, активация будет удалена. Активация может также быть удалена командой или действием другого правила, которое действовало прежде и удалило условия, необходимые для активации.

Deffunctions позволяют непосредственно определять новые функции в CLIPS. Тело deffunction - ряд выражений, подобных правилам, которые исполняются CLIPS, по вызову deffunction. Возвращаемое значение deffunction - величина последнего выражения, оцененного внутри deffunction. Вызов deffunction идентичен вызову любой другой функции в CLIPS.

Например,

```
(deffunction yes-or-no-p (?question)
  (bind ?response (ask-question ?question yes no y n))
  (if (or (eq ?response yes) (eq ?response y))
    then TRUE
    else FALSE))
```

Данная функция позволяет определить на задаваемый вопрос ответ yes или y для утверждения, no или n для отрицания.

Заключение

Язык CLIPS можно использовать для построения экспертных систем, которые смогут заменить человека при принятии решений, основанных на логическом выводе, например, при оценке «пользовательских свойств» прототипа системы. В данный момент создан прототип экспертной системы оценки пользовательского интерфейса, содержащей 25 правил, 5 функций. Благодаря системе можно проводить тестирование прототипа программного обеспечения, предоставляются рекомендации при проектировании пользовательского интерфейса. С помощью системы проводится оценка прототипа автоматизированной системы «Контингент-Сессия», позволяющей снизить трудозатраты, связанные с документооборотом, сократить время на поиск необходимой информации и формирование необходимых отчетов.